CS 343: Artificial Intelligence

Bayes Nets: Inference

Prof. Scott Niekum — The University of Texas at Austin

[These slides based on those of Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]
Bayes Net Representation

- A directed, acyclic graph, one node per random variable
- A conditional probability table (CPT) for each node
  - A collection of distributions over $X$, one for each combination of parents’ values

$$P(X|a_1 \ldots a_n)$$

- Bayes’ nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i))$$
Example: Alarm Network

<table>
<thead>
<tr>
<th>B</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+b</td>
<td>0.001</td>
</tr>
<tr>
<td>-b</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>P(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+e</td>
<td>0.002</td>
</tr>
<tr>
<td>-e</td>
<td>0.998</td>
</tr>
</tbody>
</table>

| B | E | A | P(A|B,E) |
|---|---|---|---------|
| +b| +e| +a| 0.95    |
| +b| +e| -a| 0.05    |
| +b| -e| +a| 0.94    |
| +b| -e| -a| 0.06    |
| -b| +e| +a| 0.29    |
| -b| +e| -a| 0.71    |
| -b| -e| +a| 0.001   |
| -b| -e| -a| 0.999   |

### Alarm Network

- **Burglary** (B)
- **Earthquake** (E)
- **Alarm** (A)
- **John** calls (J)
- **Mary** calls (M)

| A  | J    | P(J|A) |
|----|------|------|
| +a | +j   | 0.9  |
| +a | -j   | 0.1  |
| -a | +j   | 0.05 |
| -a | -j   | 0.95 |

| A  | M    | P(M|A) |
|----|------|-------|
| +a | +m   | 0.7   |
| +a | -m   | 0.3   |
| -a | +m   | 0.01  |
| -a | -m   | 0.99  |
Example: Alarm Network

\[
P(+b, -e, +a, -j, +m) = P(+b)P(-e|+b)P(+a|+b, -e)P(-j|a)P(+m|a) =
\]

<table>
<thead>
<tr>
<th>B</th>
<th>P(B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+b</td>
<td>0.001</td>
</tr>
<tr>
<td>-b</td>
<td>0.999</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>E</th>
<th>P(E)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+e</td>
<td>0.002</td>
</tr>
<tr>
<td>-e</td>
<td>0.998</td>
</tr>
</tbody>
</table>

| A   | J   | P(J|A) |
|-----|-----|-------|
| +a  | +j  | 0.9   |
| +a  | -j  | 0.1   |
| -a  | +j  | 0.05  |
| -a  | -j  | 0.95  |

| A   | M   | P(M|A) |
|-----|-----|-------|
| +a  | +m  | 0.7   |
| +a  | -m  | 0.3   |
| -a  | +m  | 0.01  |
| -a  | -m  | 0.99  |

| B   | E   | A   | P(A|B,E) |
|-----|-----|-----|---------|
| +b  | +e  | +a  | 0.95    |
| +b  | +e  | -a  | 0.05    |
| +b  | -e  | +a  | 0.94    |
| +b  | -e  | -a  | 0.06    |
| -b  | +e  | +a  | 0.29    |
| -b  | +e  | -a  | 0.71    |
| -b  | -e  | +a  | 0.001   |
| -b  | -e  | -a  | 0.999   |
Example: Alarm Network

\[
P(+b, -e, +a, -j, +m) = \]
\[
P(+b)P(-e)P(+a| +b, -e)P(-j| +a)P(+m| +a) = \]
\[
0.001 \times 0.998 \times 0.94 \times 0.1 \times 0.7
\]
D-Separation

- **Question:** Are $X$ and $Y$ conditionally independent given evidence variables $\{Z\}$?
  - Yes, if $X$ and $Y$ "d-separated" by $Z$
  - Consider all (undirected) paths from $X$ to $Y$
  - No active paths = independence!

- **A path is active if each triple is active:**
  - Causal chain $A \rightarrow B \rightarrow C$ where $B$ is unobserved (either direction)
  - Common cause $A \leftarrow B \rightarrow C$ where $B$ is unobserved
  - Common effect (aka v-structure)
    - $A \rightarrow B \leftarrow C$ where $B$ or one of its descendants is observed

- **All it takes to block a path is a single inactive segment**
Bayes Nets

- Representation
- Conditional Independences
  - Probabilistic Inference
    - Enumeration (exact, exponential complexity)
    - Variable elimination (exact, worst-case exponential complexity, often better)
    - Inference is NP-complete
    - Sampling (approximate)
  - Learning Bayes Nets from Data
Inference

- Inference: calculating some useful quantity from a joint probability distribution

- Examples:
  - Posterior probability
    \[ P(Q|E_1 = e_1, \ldots E_k = e_k) \]
  - Most likely explanation:
    \[ \arg\max_q P(Q = q|E_1 = e_1 \ldots) \]
Inference by Enumeration

- **General case:**
  - **Evidence variables:** \( E_1 \ldots E_k = e_1 \ldots e_k \)
  - **Query** variable: \( Q \)
  - **Hidden variables:** \( H_1 \ldots H_r \)
  \[ \left\{ X_1, X_2, \ldots X_n \right\} \]

- **We want:**
  \[ P(Q|e_1 \ldots e_k) \]

- **Step 1:** Select the entries consistent with the evidence

- **Step 2:** Sum out H to get joint of Query and evidence

- **Step 3:** Normalize

\[ Z = \sum_q P(Q, e_1 \ldots e_k) \]

\[ P(Q|e_1 \ldots e_k) = \frac{1}{Z} P(Q, e_1 \ldots e_k) \]
Inference by Enumeration in Bayes Net

- Given unlimited time, inference in BNs is easy
- Reminder of inference by enumeration by example:

\[
P(B \mid +j, +m) \propto P(B, +j, +m) \\
= \sum_{e, a} P(B, e, a, +j, +m) \\
= \sum_{e, a} P(B)P(e)P(a \mid B, e)P(+j \mid a)P(+m \mid a)
\]

\[
= P(B)P(+e)P(+a \mid B, +e)P(+j + a)P(+m + a) + P(B)P(+e)P(-a \mid B, +e)P(+j - a)P(+m - a) \\
+ P(B)P(-e)P(+a \mid B, -e)P(+j + a)P(+m + a) + P(B)P(-e)P(-a \mid B, -e)P(+j - a)P(+m - a)
\]
Inference by Enumeration?

\[ P(Antilock|\text{observed variables}) = ? \]
Why is inference by enumeration so slow?
- You join up the whole joint distribution before you sum out the hidden variables

Idea: interleave joining and marginalizing!
- Called “Variable Elimination”
- Still NP-hard, but usually much faster than inference by enumeration

First we’ll need some new notation: factors
Factor Zoo
Factor Zoo I

- **Joint distribution**: $P(X,Y)$
  - Entries $P(x,y)$ for all $x, y$
  - Sums to 1

- **Selected joint**: $P(x,Y)$
  - A slice of the joint distribution
  - Entries $P(x,y)$ for fixed $x$, all $y$
  - Sums to $P(x)$

- **Number of capitals** = dimensionality of the table
- **Single conditional:** $P(Y | x)$
  - Entries $P(y | x)$ for fixed $x$, all $y$
  - Sums to 1

- **Family of conditionals:**
  - $P(Y | X)$
    - Multiple conditionals
    - Entries $P(y | x)$ for all $x, y$
    - Sums to $|X|$
- Specified family: $P(\ y \mid X\ )$
  - Entries $P(y \mid x)$ for fixed $y$, but for all $x$
  - Sums to ... who knows!

$$P(\text{rain} \mid T)$$

<table>
<thead>
<tr>
<th>T</th>
<th>W</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>rain</td>
<td>0.2</td>
</tr>
<tr>
<td>cold</td>
<td>rain</td>
<td>0.6</td>
</tr>
</tbody>
</table>
In general, when we write $P(Y_1 \ldots Y_N \mid X_1 \ldots X_M)$

- It is a “factor,” a multi-dimensional array
- Its values are $P(y_1 \ldots y_N \mid x_1 \ldots x_M)$
- Any assigned (=lower-case) $X$ or $Y$ is a dimension missing (selected) from the array
Example: Traffic Domain

- **Random Variables**
  - R: Raining
  - T: Traffic
  - L: Late for class!

\[
P(L) = ? = \sum_{r,t} P(r,t,L) = \sum_{r,t} P(r)P(t|r)P(L|t)
\]
Inference by Enumeration: Procedural Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)

\[ P(R) \]
\[
\begin{array}{c|c}
+r & 0.1 \\
-r & 0.9 \\
\end{array}
\]

\[ P(T|R) \]
\[
\begin{array}{c|c|c}
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\end{array}
\]

\[ P(L|T) \]
\[
\begin{array}{c|c|c}
+t & +l & 0.3 \\
+t & -l & 0.7 \\
-t & +l & 0.1 \\
-t & -l & 0.9 \\
\end{array}
\]

- Any known values are selected
  - E.g. if we know \( L = +l \) then the initial factors are:

\[ P(R) \]
\[
\begin{array}{c|c}
+r & 0.1 \\
-r & 0.9 \\
\end{array}
\]

\[ P(T|R) \]
\[
\begin{array}{c|c|c}
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\end{array}
\]

\[ P(+l|T) \]
\[
\begin{array}{c|c|c}
+t & +l & 0.3 \\
-t & +l & 0.1 \\
\end{array}
\]

- Procedure: Join all factors, then eliminate all hidden variables
Operation 1: Join Factors

- First basic operation: joining factors
- Combining factors:
  - Get all factors over the joining variable
  - Build a new factor over the union of the variables involved
- Example: Join on R

\[
P(R) \times P(T|R) \rightarrow P(R, T)
\]

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>-r</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>+t</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>+t</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>-t</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

\[\forall r, t : P(r, t) = P(r) \cdot P(t|r)\]
Example: Multiple Joins
Example: Multiple Joins

\[
P(R) \begin{array}{c|c}
+ r & 0.1 \\
- r & 0.9 \\
\end{array}
\]

\[
P(T|R) \begin{array}{c|c|c}
+ r & + t & 0.8 \\
+ t & - t & 0.2 \\
- r & + t & 0.1 \\
- r & - t & 0.9 \\
\end{array}
\]

\[
P(L|T) \begin{array}{c|c|c}
+ t & + l & 0.3 \\
+ t & - l & 0.7 \\
- t & + l & 0.1 \\
- t & - l & 0.9 \\
\end{array}
\]

\[
P(R,T) \begin{array}{c|c|c}
+ r & + t & 0.08 \\
+ r & - t & 0.02 \\
- r & + t & 0.09 \\
- r & - t & 0.81 \\
\end{array}
\]

\[
P(R,T,L) \begin{array}{c|c|c|c}
+ r & + t & + l & 0.024 \\
+ r & + t & - l & 0.056 \\
+ r & - t & + l & 0.002 \\
+ r & - t & - l & 0.018 \\
- r & + t & + l & 0.027 \\
- r & + t & - l & 0.063 \\
- r & - t & + l & 0.081 \\
- r & - t & - l & 0.729 \\
\end{array}
\]
Operation 2: Eliminate

- Second basic operation: **marginalization**
- Take a factor and sum out a variable
  - Shrinks a factor to a smaller one
  - A projection operation
- Example:

\[
P(R, T) \quad \text{sum } R \quad P(T)
\]

<table>
<thead>
<tr>
<th></th>
<th>+r</th>
<th>-r</th>
<th>+t</th>
<th>-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.08</td>
<td>0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-r</td>
<td>0.09</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{array}{cc}
+t & 0.17 \\
-t & 0.83 \\
\end{array}
\]
Multiple Elimination

\[ P(R, T, L) \]

\[
\begin{array}{ccc}
+r & +t & +l & 0.024 \\
+r & +t & -l & 0.056 \\
+r & -t & +l & 0.002 \\
+r & -t & -l & 0.018 \\
-r & +t & +l & 0.027 \\
-r & +t & -l & 0.063 \\
-r & -t & +l & 0.081 \\
-r & -t & -l & 0.729 \\
\end{array}
\]

Sum out R

\[ P(T, L) \]

\[
\begin{array}{cc}
+t & +l & 0.051 \\
+t & -l & 0.119 \\
-t & +l & 0.083 \\
-t & -l & 0.747 \\
\end{array}
\]

Sum out T

\[ P(L) \]

\[
\begin{array}{c}
+l & 0.134 \\
-l & 0.886 \\
\end{array}
\]
Thus Far: Multiple Join, Multiple Eliminate (= Inference by Enumeration)
Marginalizing Early (= Variable Elimination)
Traffic Domain

\[ P(L) = ? \]

- **Inference by Enumeration**

\[ \sum_t \sum_r P(L|t)P(r)P(t|r) \]

- **Variable Elimination**

\[ \sum_t P(L|t) \sum_r P(r)P(t|r) \]

- Join on \( r \)

- Join on \( t \)

- Eliminate \( r \)

- Eliminate \( t \)
Marginalizing Early! (aka VE)

\[
P(R)
\]

<table>
<thead>
<tr>
<th>r</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[
P(T|R)
\]

<table>
<thead>
<tr>
<th>r</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.8</td>
</tr>
<tr>
<td>+r</td>
<td>0.2</td>
</tr>
<tr>
<td>-r</td>
<td>0.1</td>
</tr>
<tr>
<td>-r</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[
P(R, T)
\]

<table>
<thead>
<tr>
<th>r</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>0.08</td>
</tr>
<tr>
<td>+r</td>
<td>0.02</td>
</tr>
<tr>
<td>-r</td>
<td>0.09</td>
</tr>
<tr>
<td>-r</td>
<td>0.81</td>
</tr>
</tbody>
</table>

\[
P(T)
\]

<table>
<thead>
<tr>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
</tr>
<tr>
<td>-t</td>
</tr>
</tbody>
</table>

\[
P(L|R, T)
\]

<table>
<thead>
<tr>
<th>t</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>0.3</td>
</tr>
<tr>
<td>+t</td>
<td>0.7</td>
</tr>
<tr>
<td>-t</td>
<td>0.1</td>
</tr>
<tr>
<td>-t</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[
P(L|T)
\]

<table>
<thead>
<tr>
<th>t</th>
<th>l</th>
</tr>
</thead>
<tbody>
<tr>
<td>+t</td>
<td>0.3</td>
</tr>
<tr>
<td>+t</td>
<td>0.7</td>
</tr>
<tr>
<td>-t</td>
<td>0.1</td>
</tr>
<tr>
<td>-t</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[
P(L)
\]

<table>
<thead>
<tr>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>+l</td>
</tr>
<tr>
<td>-l</td>
</tr>
</tbody>
</table>
If evidence, start with factors that select that evidence

- No evidence uses these initial factors:

\[
\begin{array}{c|c}
P(R) & P(T|R) & P(L|T) \\
\hline
+r & +t & 0.8 \\
+r & -t & 0.2 \\
-r & +t & 0.1 \\
-r & -t & 0.9 \\
\end{array}
\]

- Computing \( P(L|+r) \), the initial factors become:

\[
\begin{array}{c|c}
P(+r) & P(T|+r) & P(L|T) \\
\hline
+r & +t & 0.8 \\
+r & -t & 0.2 \\
\hline
\end{array}
\]

- We eliminate all vars other than query + evidence
Result will be a selected joint of query and evidence

- E.g. for $P(L \mid +r)$, we would end up with:

$P(+r, L)$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+r</td>
<td>+l</td>
<td>0.026</td>
</tr>
<tr>
<td>+r</td>
<td>-l</td>
<td>0.074</td>
</tr>
</tbody>
</table>

Normalize $P(L \mid +r)$

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+l</td>
<td>0.26</td>
</tr>
<tr>
<td>-l</td>
<td>0.74</td>
</tr>
</tbody>
</table>

To get our answer, just normalize this!

That’s it!
General Variable Elimination

- **Query:** \( P(Q|E_1 = e_1, \ldots, E_k = e_k) \)

- **Start with initial factors:**
  - Local CPTs (but instantiated by evidence)

- **While there are still hidden variables (not Q or evidence):**
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H

- **Join all remaining factors and normalize**
Example

\[ P(B \mid j, m) \propto P(B, j, m) \]

\[
\begin{array}{cccccc}
P(B) & P(E) & P(A \mid B, E) & P(j \mid A) & P(m \mid A) \\
\end{array}
\]

Choose A

\[
\begin{align*}
P(A \mid B, E) & \quad \times \quad P(j, m, A \mid B, E) & \quad \sum \quad P(j, m \mid B, E) \\
P(j \mid A) & \quad \times \quad P(j, m, A \mid B, E) & \quad \sum \quad P(j, m \mid B, E) \\
P(m \mid A) & \quad \times \quad P(j, m, A \mid B, E) & \quad \sum \quad P(j, m \mid B, E) \\
\end{align*}
\]

\[
\begin{array}{cccc}
P(B) & P(E) & P(j, m \mid B, E) \\
\end{array}
\]
Choose E

\[
P(E) \times P(j, m, E | B) \sum P(j, m | B)
\]

Finish with B

\[
P(B) \times P(j, m, B) \text{ Normalize } P(B | j, m)
\]
Same Example in Equations

\[ P(B|j,m) \propto P(B,j,m) \]

\[
\begin{array}{cccccc}
P(B) & P(E) & P(A|B,E) & P(j|A) & P(m|A) \\
\end{array}
\]

\[
P(B|j,m) \propto P(B,j,m) \\
= \sum_{e,a} P(B,j,m,e,a) \\
= \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) \\
= \sum_{e} P(B)P(e) \sum_{a} P(a|B,e)P(j|a)P(m|a) \\
= \sum_{e} P(B)P(e)f_1(B,e,j,m) \\
= P(B) \sum_{e} P(e)f_1(B,e,j,m) \\
= P(B)f_2(B,j,m)
\]

marginal can be obtained from joint by summing out

use Bayes’ net joint distribution expression

use \( x^*(y+z) = xy + xz \)

joining on a, and then summing out gives \( f_1 \)

joining on e, and then summing out gives \( f_2 \)
For the query $P(X_n \mid y_1, \ldots, y_n)$ work through the following two different orderings: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. What is the size of the maximum factor generated for each of the orderings?

- **Answer:** $2^{n+1}$ versus $2^2$ (assuming binary)

- **In general:** the ordering can greatly affect efficiency.
The computational and space complexity of variable elimination is determined by the largest factor.

The elimination ordering can greatly affect the size of the largest factor.
- E.g., previous slide’s example $2^n$ vs. 2

Does there always exist an ordering that only results in small factors?
- No!
Worst Case Complexity?

- **CSP:**
  \[
  (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_3 \lor \neg x_4) \land (x_2 \lor \neg x_2 \lor \neg x_4) \land (\neg x_3 \lor x_4 \lor \neg x_5) \land (x_2 \lor x_5 \lor x_7) \land (x_4 \lor x_5 \lor x_6) \land (\neg x_5 \lor x_6 \lor \neg x_7) \land (\neg x_5 \lor \neg x_6 \lor x_7)
  \]

- \[P(X_i = 0) = P(X_i = 1) = 0.5\]

- \[Y_1 = X_1 \lor X_2 \lor \neg X_3\]

- \[Y_8 = \neg X_5 \lor X_6 \lor X_7\]

- \[Y_{1,2} = Y_1 \land Y_2\]

- \[Y_{7,8} = Y_7 \land Y_8\]

- \[Y_{1,2,3,4} = Y_{1,2} \land Y_{3,4}\]

- \[Y_{5,6,7,8} = Y_{5,6} \land Y_{7,8}\]

- \[Z = Y_{1,2,3,4} \land Y_{5,6,7,8}\]

- If we can answer \(P(z)\) equal to zero or not, we answered whether the 3-SAT problem has a solution.

- Hence inference in Bayes nets is NP-hard. No known efficient probabilistic inference in general.
A polytree is a directed graph with no undirected cycles

For poly-trees you can always find an ordering that is efficient
  ▪ Try it!!
  ▪ Very similar to tree-structured CSP algorithm

Cut-set conditioning for Bayes net inference
  ▪ Choose set of variables such that if removed only a polytree remains
  ▪ Exercise: Think about how the specifics would work out!
Bayes Nets

- Representation
- Conditional Independences
  - Probabilistic Inference
    - Enumeration (exact, exponential complexity)
    - Variable elimination (exact, worst-case exponential complexity, often better)
  - Inference is NP-complete
    - Sampling (approximate)
  - Learning Bayes Nets from Data